COMP90042

Web search and text analysis

Workshop Week 8

Your tutor

- Winn Chow (Senior Tutor)
- winn.chow1@unimelb.edu.au
- Office: Doug McDonell - 9.23
- Here, you can find my workshop slides:
- https://github.com/winnchow/COMP90042-Workshops

Q1

1. What is **Information Extraction**? What might the "extracted" information look like?
   (a) What is **Named Entity Recognition** and why is it difficult? What might make it more difficult for persons rather than places, and *vice versa*?
   (b) What is the **IOB** trick, in a sequence labelling context? Why is it important?
   (c) What is **Relation Extraction**? How is it similar to NER, and how is it different?
   (d) Why are hand–written patterns generally inadequate for IE, and what other approaches can we take?

Q1

– **Information Extraction**

– We want to extract information from a (generally unstructured) document, into a structured format that we can sensibly query later.

- Given this:
  - “Brasilia, the Brazilian capital, was founded in 1960.”
- Obtain this:
  - capital(Brazil, Brasilia)
  - founded(Brasilia, 1960)

Named entity recognition

Citing high fuel prices, **[ORG United Airlines]** said **[TIME Friday]** it has increased fares by **[MONEY $6]** per round trip on flights to some cities also served by lower-cost carriers. **[ORG American Airlines]**, a unit of **[ORG AMR Corp.]**, immediately matched the move, spokesman **[PER Tim Wagner]** said. **[ORG United]**, a unit of **[ORG UAL Corp.]**, said the increase took effect **[TIME Thursday]** and applies to most routes where it competes against discount carriers, such as **[LOC Chicago]** to **[LOC Dallas]** and **[LOC Denver]** to **[LOC San Francisco]**

People vs. Place

- One common problem, that we see with both people’s names and places, is that they are **ambiguous with common nouns**.
- Generally speaking, we can write a (somewhat) **exhaustive list of names of places — a gazetteer** —but we can’t with names of people, which are constantly changing.
- On the other hand, **many different locations can have the same name (e.g. Melbourne, Australia and Melbourne, USA).**

Dealing with adjacent entities: IOB tagging

- **[ORG American Airlines],** a unit of **[ORG AMR Corp.]**, immediately matched the move, spokesman **[PER Tim Wagner]** said.

- American/**B-ORG** Airlines**/I-ORG** ,/**O** a/**O** unit/**O** of/**O** AMR/**B-ORG** Corp./**I-ORG** ,/**O** immediately/**O** matched/**O** the/**O** move/**O** , /**O** spokesman/**O** Tim/**B-PER** Wagner/**I-PER** said/**O** ./**O**

- **B-ORG** represents the *beginning* of an **ORG** entity. If the entity has more than one token, subsequent tags are represented as **I-ORG**.

Relation extraction - methods

- If we have access to a fixed relation database:
  * Rule-based
  * Supervised
  * Semi-supervised
  * Distant supervision
- If no restrictions on relations:
  * Unsupervised
  * Sometimes referred as “OpenIE”

Supervised relation extraction

- **[ORG American Airlines],** a unit of **[ORG AMR Corp.]**, immediately matched the move, spokesman **[PER Tim Wagner]** said.
- First:
  - (American Airlines, AMR Corp.) -> positive
  - (Tim Wagner, American Airlines) -> positive
  - (Tim Wagner, AMR Corp.) -> negative
- Second:
  - (American Airlines, AMR Corp.) -> subsidiary
  - (Tim Wagner, American Airlines) -> employment

Semi-supervised relation extraction

- Annotated corpora is very expensive to create.
- Assume we have a small set of **seed tuples**.
- Mine the web for text containing the tuples:
  - Given hub(Ryanair, Charleroi)
  - Get sentences containing all terms, e.g.,
    "Budget airline **Ryanair**, which uses **Charleroi** as a **hub**, scrapped all weekend flights out of the airport."
  - Use these patterns to new tuples, e.g., hub(Jetstar, Avalon) as these words occur in similar contexts; repeat
- Suffers from "semantic drift", where errors compound

Distant supervision

- Semi-supervised methods assume the existence of seed tuples.
- What about mining new tuples?
- Distant supervision obtain new tuples from a range of sources:
  - DBpedia
  - Freebase
- Generate massive training sets, enabling the use of richer features, and no risk of semantic drift
- Still rely on a fixed set of relations.

ReVERB: Unsupervised relation extraction

- If there is no relation database or the goal is to find new relations, unsupervised approaches must be used.
- Relations become substrings, usually containing a verb
- "United has a hub in Chicago, which is the headquarters of United Continental Holdings."
  - "has a hub in"(United, Chicago)
  - "is the headquarters of"(Chicago, United Continental Holdings)
- Main problem: mapping the substring relations into canonical forms

Q2

2. What is **Question Answering**, and how is it related to **Information Retrieval** and Information Extraction?

   (a) What is **semantic parsing**, and why might it be desirable for QA? Why might approaches like NER be more desirable?

   (b) What might be the main steps for answering a question for a QA system?

Q2a

- **Semantic Parsing**
  - To define the (meaning–based) relations between those elements.
- `Donald Trump is president of the United States.`
  - We might be trying to generate a logical relationship like `is(Donald Trump, president(United States)).`

Semantic Parsing

- Based on aligned questions and their logical form, e.g., GeoQuery (Zelle & Mooney 1996)

  What is the capital of the state with the largest population?

  answer(C, (capital(S,C), largest(P, (state(S), population(S,P))))).

- Can model using parsing (Zettlemoyer & Collins 2005) to build compositional logical form

$$
\begin{array}{cccc}
\text{What} & \text{states} & \text{border} & \text{Texas} \\
\hline
(S/(S\backslash NP))/N & N & (S\backslash NP)/NP & NP \\
\lambda f.\lambda g.\lambda x.f(x) \wedge g(x) & \lambda x.state(x) & \lambda x.\lambda y.borders(y,x) & texas \\
\hline
\multicolumn{2}{c}{S/(S\backslash NP) \quad >} & \multicolumn{2}{c}{(S\backslash NP) \quad >} \\
\multicolumn{2}{c}{\lambda g.\lambda x.state(x) \wedge g(x)} & \multicolumn{2}{c}{\lambda y.borders(y,texas)} \\
\hline
\multicolumn{4}{c}{S \quad >} \\
\multicolumn{4}{c}{\lambda x.state(x) \wedge borders(x,texas)}
\end{array}
$$

IR-based Factoid QA: TREC-QA

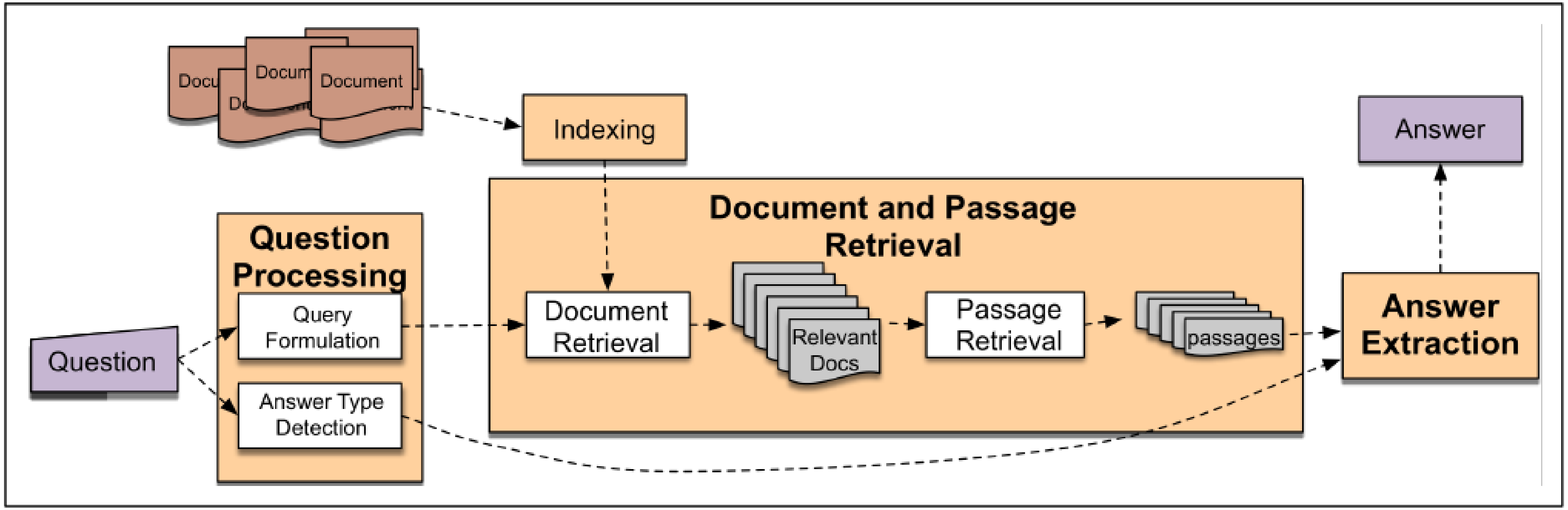

Figure 23.2 IR-based factoid question answering has three stages: question processing, passage retrieval, and answer processing.

1. Use question to make query for IR engine
2. Find document, and passage within document
3. Extract short answer string

QA over structured KB

- Many large knowledge bases
  - Sports statistics, Moon rock data, …
  - Freebase, DBpedia, Yago, …
- Each with own query language SQL, SPARQL etc.
- Can we support natural language queries?
  - E.g.,

"When was Ada Lovelace born?" $\rightarrow$ `birth-year (Ada Lovelace, ?x)`
"What is the capital of England?" $\rightarrow$ `capital-city(?x, England)`

  - Answer by processing query against KB; i.e., find RDF triple (Ada Lovelace, birth-year, 1815) to provide answer = 1815.

- In a Relation Extraction sense:
  - Offline, we process our document collection to generate a list of relations (our knowledge base)
  - When we receive a (textual) query, we transform it into the same structural representation, with some known field(s) and some missing field(s)
  - We examine our knowledge base for facts that match the known fields
  - We rephrase the query as an answer with the missing field(s) filled in from the matching facts from the knowledge base
- In an Information Retrieval sense:
  - Offline, we process our document collection into a suitable format for IR querying (e.g. inverted index)
  - When we receive a (textual) query, we remove irrelevant terms, and (possibly) expand the query with related terms
  - We select the best document(s) from the collection based on our querying model (e.g. TF-IDF with cosine similarity)
  - We identify one or more snippets from the best document(s) that match the query terms, to form an answer